Parte 1

1. ¿Qué es Java y cuál es su importancia en el desarrollo de software?

Es un lenguaje de programación orientado a objetos multiplaforma, es decir, puede ejectuarse en cualquier sistema operativo a través de un máquina virtual java. Cuenta con una de las comunidades que facilitan el desarrollo de nuevas librerias y frameworks , actualmente mantiene una liberación de nuevos releases cada 6 meses, lo que permite que el lenguaje pueda seguir evolucionando. Es un lenguaje ampliamente soportado por los provedores de Nube , y por su versatilidad, es utilizados en la construcción de Apis para multiples propositos y tecnologias.

2. ¿Qué es una clase en Java y cuál es su relación con los objetos?

Una clase es un boceto que permite la representación de una entidad ; abstrae las caracteristicas de una entidad del mundo real mediante la declaración la clase y sus los elementos como pueden ser atributos , métodos , constructores. El objeto es la creacíon de una clase, usualmente , a esta acción se conoce como instanciar una clase, proceso mediante el cual se crea y se asigna una dirección en la memoria mediante el uso del constructor de la clase o mediante la inyección de dependencias a través del uso de algún framework. Ejm Car car = new Car(“2013”, “owner”)

3. ¿Qué es un método en Java y cuál es su propósito?

Un metodo encapsula un conjunto de instrucciones con un próposito en particular , esta compuesto por una firma , la cual especifica los parametros de entrada y de salida, el nombre del método y la implementación de la lógica que va ejecutar el método. El uso de método tiene diversos propositos cómo la reutilización , facilidad de lectura, separación de dependencias.

4. Explica qué es la herencia en Java y proporciona un ejemplo.

Herencia en java es un proceso mediante el cual una clase, denominada subclase puede adquirir caracteristicas y comportamiento de una la clase padre.

class Car {

private String anio;

}

class Toyota extends Car {

public Toyota() {

super();

}

}

5. ¿Qué es una interfaz en Java y cómo se diferencia de una clase abstracta?

Una intefaz y una clase abstracta

6. Describe qué es Spring Boot y cuál es su objetivo principal en el desarrollo de aplicaciones.

7. ¿Cuál es la diferencia entre una aplicación Spring Boot y una aplicación Java tradicional?

8. ¿Qué es la inyección de dependencias en Spring Boot y por qué es importante?

9. ¿Qué es un Bean en Spring y cómo se configura en una aplicación Spring Boot?

10. ¿Cuál es la función principal de Spring MVC y cómo se relaciona con Spring Boot?

Parte 2

¿Qué es Docker y cuál es su propósito en el desarrollo de aplicaciones?

Docker es una tecnolgia de containerización y virtualización que permite aislar y crear un recurso independiente y poder gestionar los recursos de un computador o servidor de manera independiente a través del demonio o servicio de docker . En el desarrollo de aplicaciones, en el contexto de aplicaciones Java permite poder construir , configurar y desplegar a través de directivas, variables y comandos que se especifican en un archivo Dockerfile

¿Cuál es la diferencia entre una imagen y un contenedor en Docker?

La imagen es un plantilla de carácter inmutable que define como se va materializar o crear un contenedor . Un contenedor es la instancia en ejecución , el cual es generado a partir de una Imagen Docker

Explica qué es un Dockerfile y cuál es su función en la creación de imágenes.

Un archivo Dockerfile hace posible poder configurar, construir y ejecutar un componente de software.

¿Qué es un registro de Docker (Docker registry) y cuál es su importancia en el ecosistema de Docker?

Docker registry es un sistema de versionamiento , almacenamiento y distribución de images de Docker . Utilizar un docker registry permite tener un control sobre las imágenes, en el contexto de aplicaciones cloud nativas, permite poder un desarrollo y delivery agil de artefactos, actuando como un repositorio compartidos de imágenes ,

¿Qué es Kubernetes y cuál es su objetivo principal en la administración de contenedores?

Kubernetes es una tecnologia que permite la orquestación de contenedores , se ha convertido en un estandar en la administración de contenedores , permite poder gestionar y desplegar aplicaciones , configurar el escalamiento de acuerdo a las necesidades, monitorear la ejecución de las aplicaciones de manera automatizada.

¿Cuál es la diferencia entre un pod y un contenedor en Kubernetes?

Un pod es una unidad

Describe qué es un servicio en Kubernetes y cómo se utiliza para exponer aplicaciones.

¿Qué es un nodo (node) en el contexto de Kubernetes y cuál es su función?

Explica qué es un manifiesto de Kubernetes (Kubernetes manifest) y cómo se utiliza para definir la configuración de una aplicación.

¿Cuál es la importancia de los volúmenes persistentes en Kubernetes y para qué se utilizan?